# Abstract

In most modern multi-tasking operating systems the part played by the process scheduler is critical in ensuring that users, at a macro level, as well as processes at the system level, are treated fairly The basic functionality of traditional operating systems has already started including real-time capabilities the effective emulation of real-time scheduling algorithms has therefore become an important issue There are several advantages to providing real-time support in a general purpose operating system like Unix – for example ease of code development, availability of a large set of development tools and portability This work evaluates the performance of decay usage process schedulers through experimentation in the contexts of fairness and support for real-time applications

We quantify the fairness of decay usage process schedulers and suggest a method to improve the fairness of the normal Unix process scheduler based on detailed experimental studies We observe that standard assumptions about the start of the decay period and the execution time of processes used in analytical studies of decay usage scheduler are unrealistic In fact, in our experiments we find that unfairness in the form of overtaking among processes, which could not happen if the assumptions were valid occurs frequently

In the second part of this work an experimental evaluation of emulating soft real-time scheduling algorithms is done by modifying the Linux operating system's source code The related literature contains simulation studies, but we find that it is extremely difficult to reproduce the workload conditions in an experimental setup As a result the simulation and experimental results do need not always agree making simulation an

unsuitable methodology to study this issue Depending on the evaluation criteria, different algorithms appear to be best, this is neatly captured by two new performance measures that we introduce Our experimental results show that the default Unix decay usage process scheduling performs comparably to the POSIX process scheduler as long as the mean execution time of tasks is smaller than the CPU quantum assigned per process